Technical Committee Review Coverage

The role of the Technical Committee in publishing versions or portions of the DDI standard for review or use includes reviewing the content in terms of its adherence to DDI modeling guidelines. Specific versions may have additional guidelines (such as the DDI-Codebook concerning the retention and non-restriction of all existing classes).

References are to the PDF documents on the page Guidelines for Business Modelers.[[1]](#footnote-1)

# Structural Package Content Review:

The purpose of this review is to verify that modeling guidelines have been followed.

|  |  |  |
| --- | --- | --- |
| **Object of Review** | **Rule** | **Guideline Reference** |
| Use of primitives and complex data types | All properties should have a datatype that is either a Primitive or a Complex Data Type  No relationship can have a Target that is a Primitive or a Complex Data Type  No property in a Class with an extension base of Identifiable should have any of the following data types (these are limited to use by other Complex Data Types):  AnnotationDate  BasedOnObject  Command  CommandFile  Context  LiteralText  LocalId  NonIsoDateType  ResourceIdentifier  String  StructuredCommand  Text  TextContent | **Class Types:** Identifiable, Annotated Identifiable, Complex Data Type  **ALERT:** when making changes to Complex Data Types |
| Use of Annotation and contents | All uses of Annotation other than its use in AnnotatedIdentifiable must have a property name that designates the object the annotation describes (i.e. citationOfExternalMaterial) | **Annotation:** Annotation and it's properties |
| Use of appropriate Cardinality | Properties:  Always 0..1 or 0..n UNLESS one of the following is true:  Default value specified  Fixed value specified  Required for class to function and the use of the class is optional  Relationships:  Source Cardinality:  Composition type: 1..1  Aggregation type: 0..1 or 0..n  Neither (Simple) type: 0..n  Target Cardinality:  0..1 to 0..n UNLESS all of the following is true:  Source class is unusable if Target is not provided  Information is available at all production points  Minimum number of Targets required (i.e. a polygon requires a minimum of 4 points to describe it)  When overwriting an inherited relationship the cardinality may NOT be relaxed | **Cardinality:** Properties and Relations |
| Use of Identifiable and AnnotatedIdentifiable | All classes not found in Primitives or Complex Data Types must extend from Identifiable at their base  All classes with Identifiable at their base must have an independent existence and have a possible relationship to multiple instances of a class or classes (i.e. CodeItem)  All classes that extend from AnnotatedIdentifiable must contain content that needs to be discoverable in its own right and expresses intellectual property in and of itself | **Class Types:** Identifiable, Annotated Identifiable, Complex Data Type |
| Consistency in Class structure: Property usage | When overwriting an inherited property the cardinality may NOT be relaxed  URN, URL, and URI:  All datatype will be xs:anyURI  Property name MUST be “uri” [lower case] UNLESS:  A URN or URL is explicitly required OR  Multiple properties of type xs:anyURI need to be differentiated by purpose or usage (i.e. uriBlueprint)  Text datatypes are limited to the following and usage must match their descriptive usage and structure notes:  DynamicText  ExternalControlledVocabularyEntry  InternationalString  OneCharString  PairedExternalControlledVocabularyEntry  InternationalStructuredString  TypedDescriptiveText  Value  xs:string  Date datatypes are limited to the following and usage must match their descriptive usage and structure notes:  Date  DateRange  IsoDate  ReferenceDate  Descriptive properties must be one of the following and contain the standard name, datatype, cardinality, and description. The description of the property may have content added to clarify its use within the specific class.  Name  DisplayLabel  Definition  Purpose  Usage  Rationale  Overview  DescriptiveText  Use of the property name “description” is NOT ALLOWED | **Property Options:** Usage  **Property Options:** Text  **Property Options:** Date  **Property Options:** Standard |
| Consistency in Class structure: pattern structures | All classes MUST be abstract  Classes should NEVER extend from AnnotatedIdentifiable  Properties and relationships should be limited to the basic requirements of the class (limit descriptive properties to those required by the role of the class)  Classes should NEVER realize another pattern class (they should use it as an extension base) | **Patterns:** Creation and Use |
| Consistency in Class structure: pattern realization | A class that “realizes” a pattern class:  MUST contain ALL of the properties and relationships of the pattern class (included those inherited from its extension base). Note that cardinality may be constrained but not relaxed.  MUST include the relationship “realizes” with Target Object=[Pattern class name], Description=”Uses pattern for [Pattern class name], Source cardinality=”0..n”, Target cardinality=”1..1”, Relationship type=”Neither”  Target Object must be constrained to a non-Pattern subtype of the pattern class  If the Pattern Class has an extension base of Identifiable (directly or inherited) the class MUST extend from Identifiable, AnnotatedIdentifiable, or any class that already realizes the same pattern class.  If the Pattern Class is a Complex Data Type (i.e. NOT Identifiable) the realization MUST be placed in ComplexDataType and may have ONLY an extension base of another ComplexDataType realizing the same pattern class OR have no extension base at all.  A class may realize more than one pattern class as long as all properties and relationships are included. The “realizes” relationship is not carried into the bindings and therefore does not get overwritten. | **Patterns:** Creation and Use |
| Consistency in Class structure: Documentation within a class | For any inherited property or relationship that is being overwritten by the inheriting class, all original documentation MUST be retained. New content may be added  Default values: Documentation MUST begin with “Default value is [value].” This may be followed with any additional documentation desired  Fixed values: Documentation MUST begin with “Fixed to [value].” This may be followed with any additional documentation desired | **Documenting Classes:** Expanding Documentation in Inheritance  **Property Options:** Default and Fixed Values |
| Verify that Design Principles are followed | Classes must conform to the design principles based on the following metrics:   1. Documentation    1. Readability    2. Acronym usage    3. Documentation reflects principles and modeling guidelines 2. Design    1. Proportion of classes that carry forward    2. Change information    3. Level of complexity – sufficient but not gratuitous       1. Balances complexity with functionality and understandability    4. Compatibility with previous versions 3. Capability    1. Coverage of existing DDI-C, DDI-L, and GSIM content | **Design Principles** |

# Functional View Content Review:

|  |  |  |
| --- | --- | --- |
| **Object of Review** | **Rule** | **Guideline Reference** |
| Content of a Functional View | DocumentInformation is included for all Functional Views intended to be persistent in nature (i.e. XML publication of a codebook for archive purposes)  MUST contain:   * ONLY classes from published packages * A clear starting point that allows for the identification and relationship information for all component classes making up the Functional View * ALL relationship targets except as noted in Functional View documentation * NO abstract classes * NO Complex Data Types * NO orphans (defined as classes unrelated to one or more entry points identified in the documentation, i.e. the contents must hold together as a unified whole) | **Functional Views:** Overview  **Functional Views:** Requirements and Structure  **Functional Views:** Step-by-Step Process |
| Documentation of a Functional View | Documentation of the Functional View MUST provide clear definition of:  Purpose  Use Case(s)  Target Audience (intended users)  List of constrained classes with documented usage  Specialized use of classes  General documentation on use of the Functional View | **Functional Views:** Requirements and Structure |

1. https://ddi-alliance.atlassian.net/wiki/spaces/DDI4/pages/37552132/Modeling+Guidelines+for+Business+Modelers [↑](#footnote-ref-1)